

FIG. 1

Application Level

130

Protocol Analysis Framework

126

Decode Management API (DLL)

128

Default Handler  
Data/Algorithm  
(DLL) 133

Protocol Specific  
Data/Algorithm  
(DLL) 132

Protocol Specific  
Data/Algorithm  
(DLL) 132

Protocol Specific  
Data/Algorithm  
(DLL) 132

Protocol Specific  
Data/Algorithm  
(DLL) 132

Decode / Filtering Engine (DLL)

134

FIG. 2

FIG. 3 is a block diagram of a system architecture for a protocol decode engine. The system includes a Decode Manager APL (128) which receives API Entry (305) and sends data to a Protocol Mux (310). The Protocol Mux (310) sends data to a Protocol Control (315) and a Multi Line Decode Cache (320). The Protocol Control (315) sends data to the Multi Line Decode Cache (320). The Multi Line Decode Cache (320) sends data to a Pointer to (325). The Pointer to (325) sends data to an AppServices Object (325) which is derived from classes that implement application specific formatting of protocol fields. The AppServices Object (325) sends data to a Protocol Specific DLL (132). The Protocol Specific DLL (132) contains Protocol Specific Data Tables (340), Protocol Configuration Dialog (345), and Protocol Filter Dialogs (350). The Protocol Specific DLL (132) sends data to a BaseProtocol DLL (134). The BaseProtocol DLL (134) contains a Protocol Decode Engine (355) and a Filtering Engine (360). The Protocol Decode Engine (355) sends data to the Filtering Engine (360).

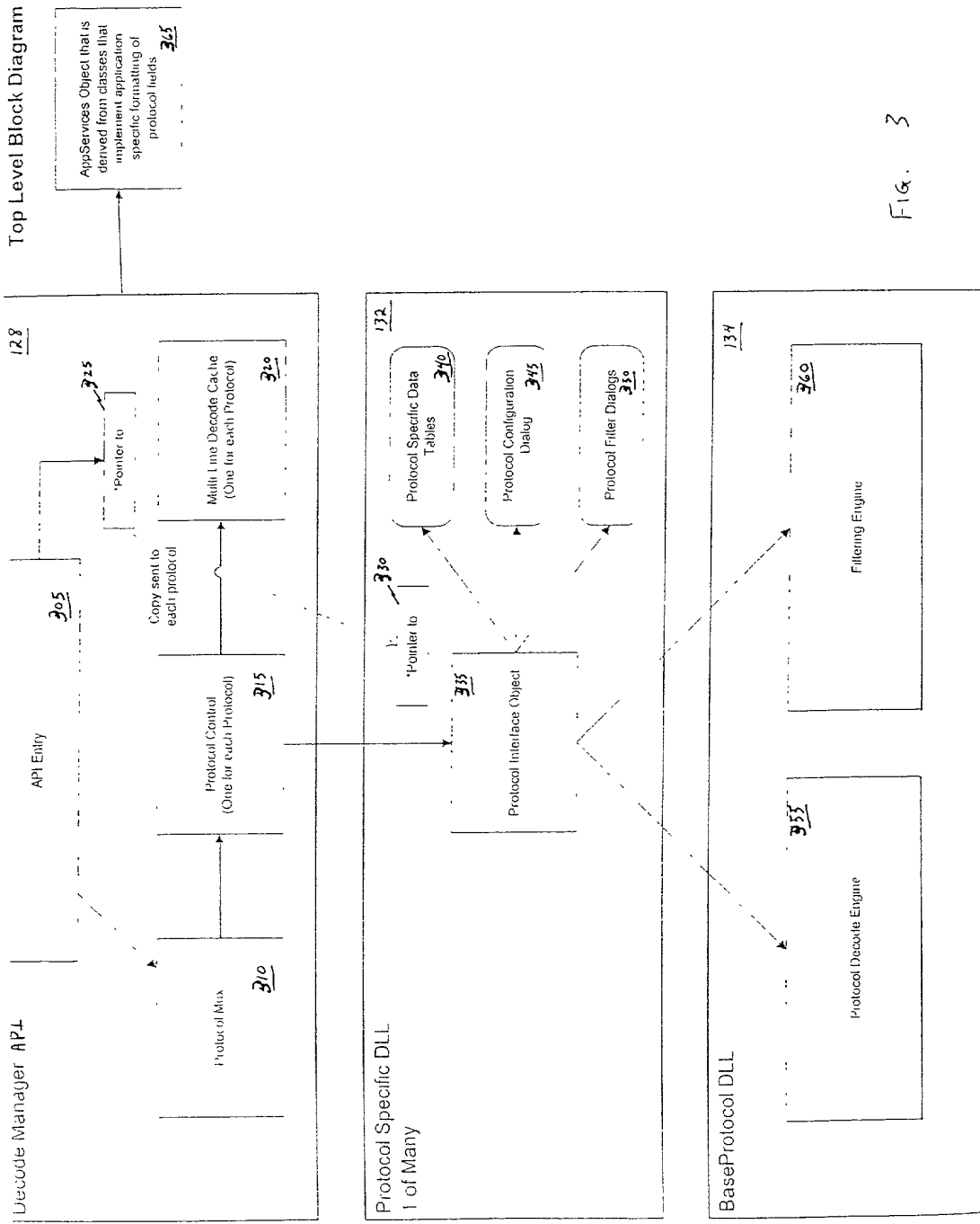


FIG. 3

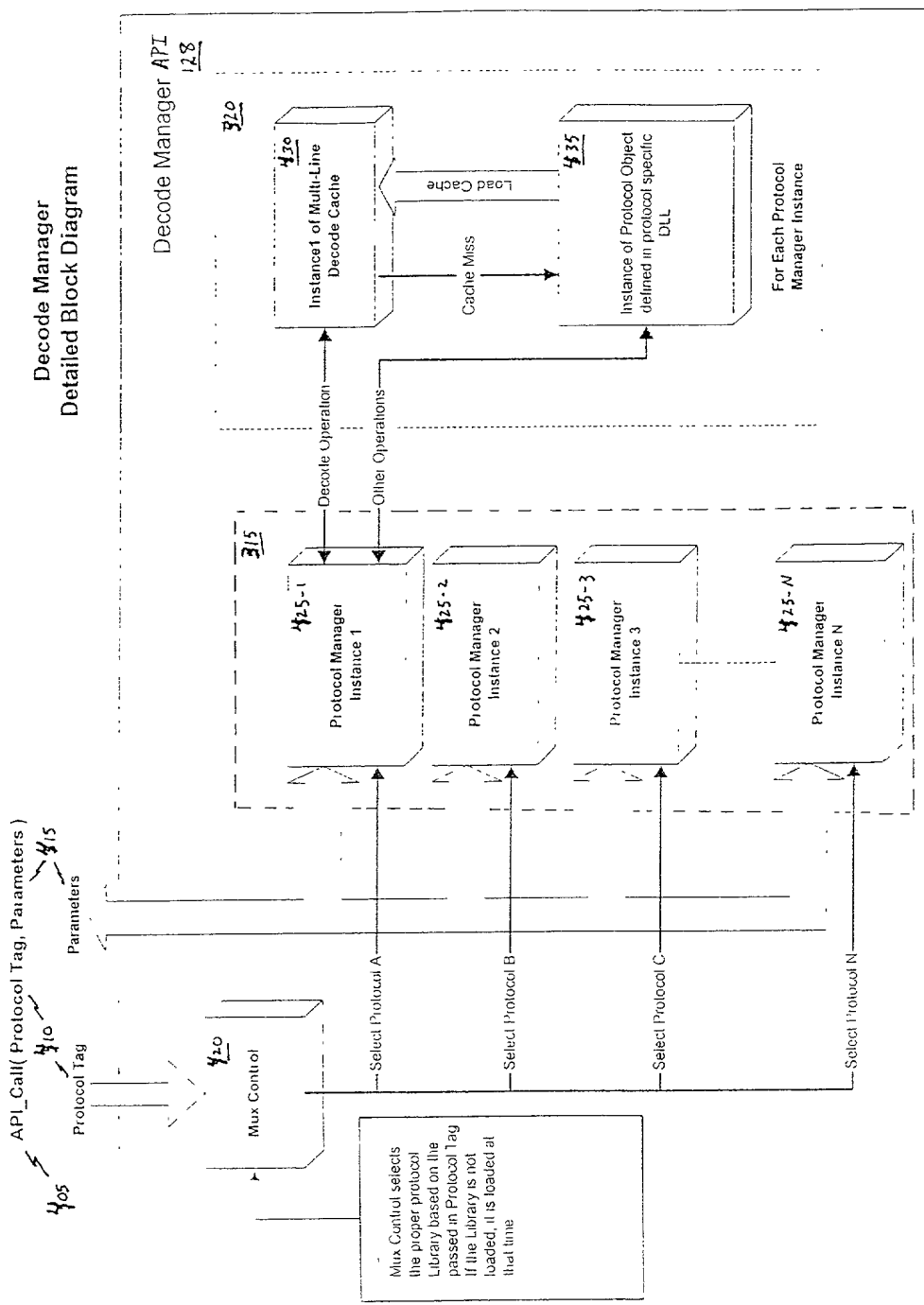


Fig 4

# API function call to library module routing algorithm flowchart

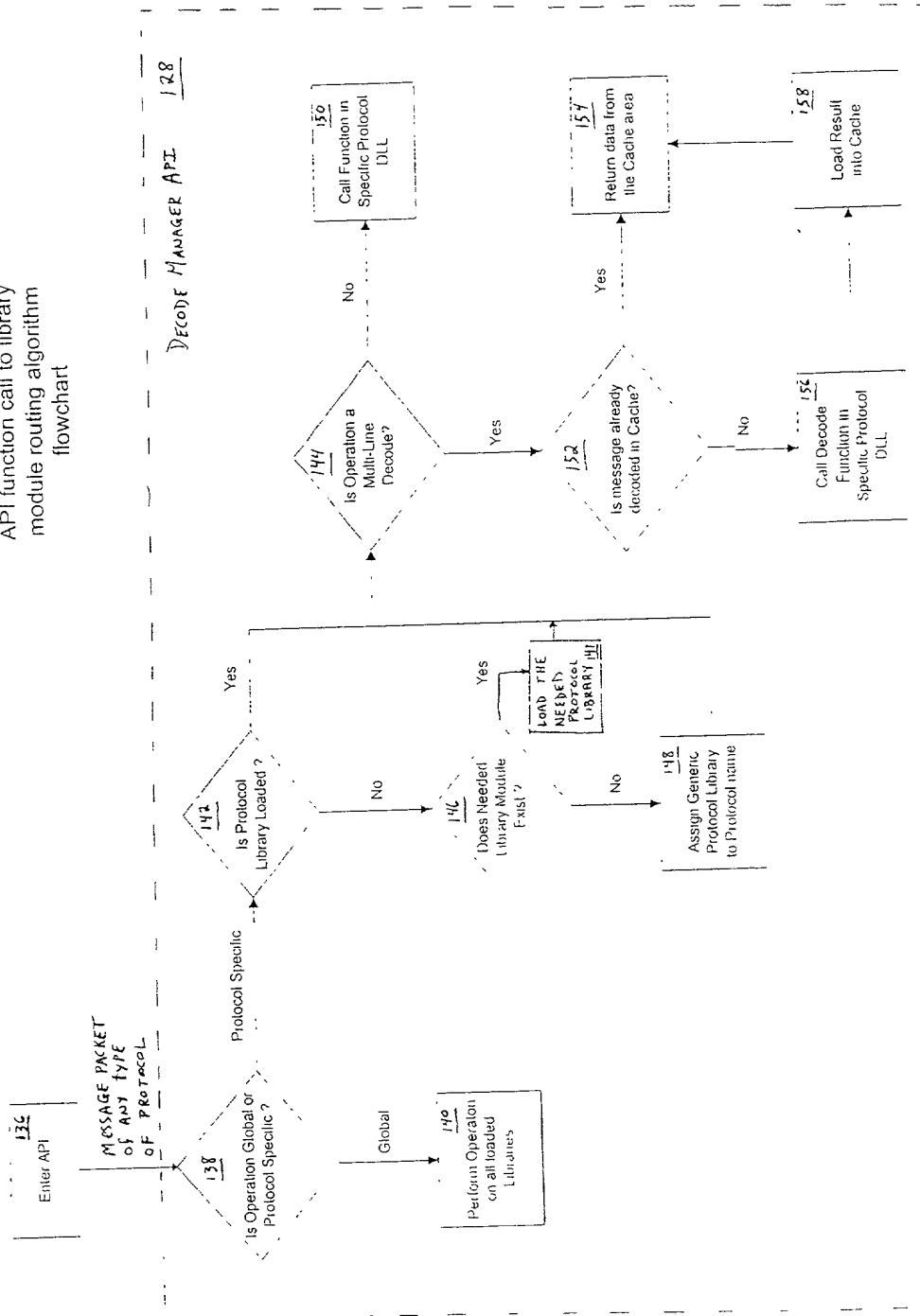


Fig 35

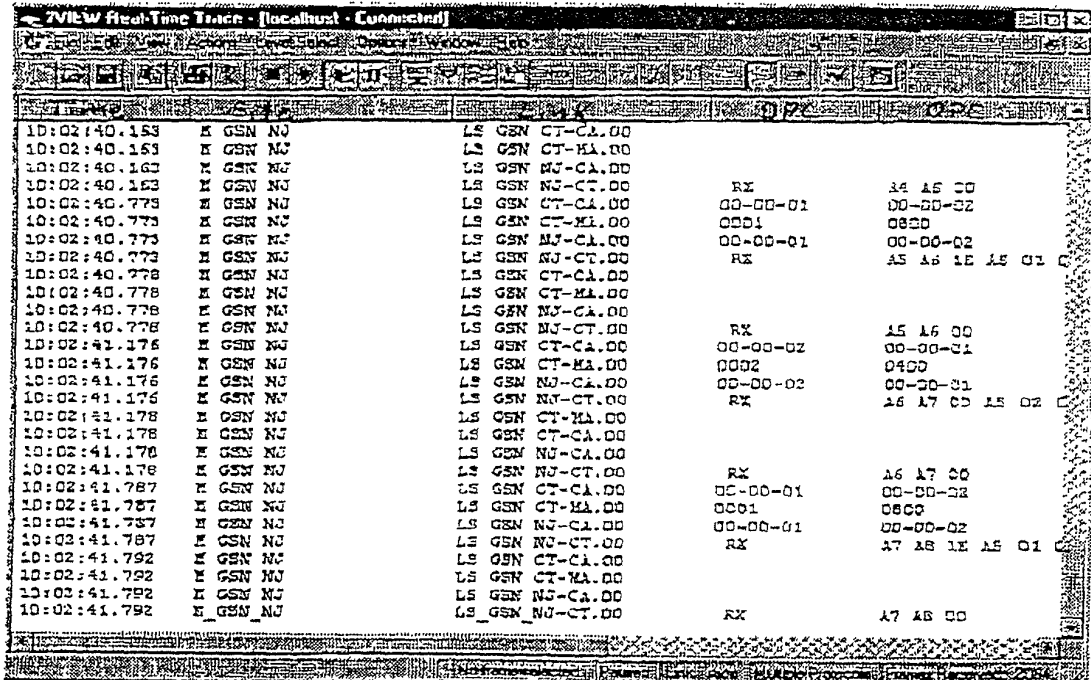


Figure 9-1. Real-Time Trace main window showing a single-line view link trace session with Layer 3 data.

FIG. 6

2025 RELEASE UNDER E.O. 14176

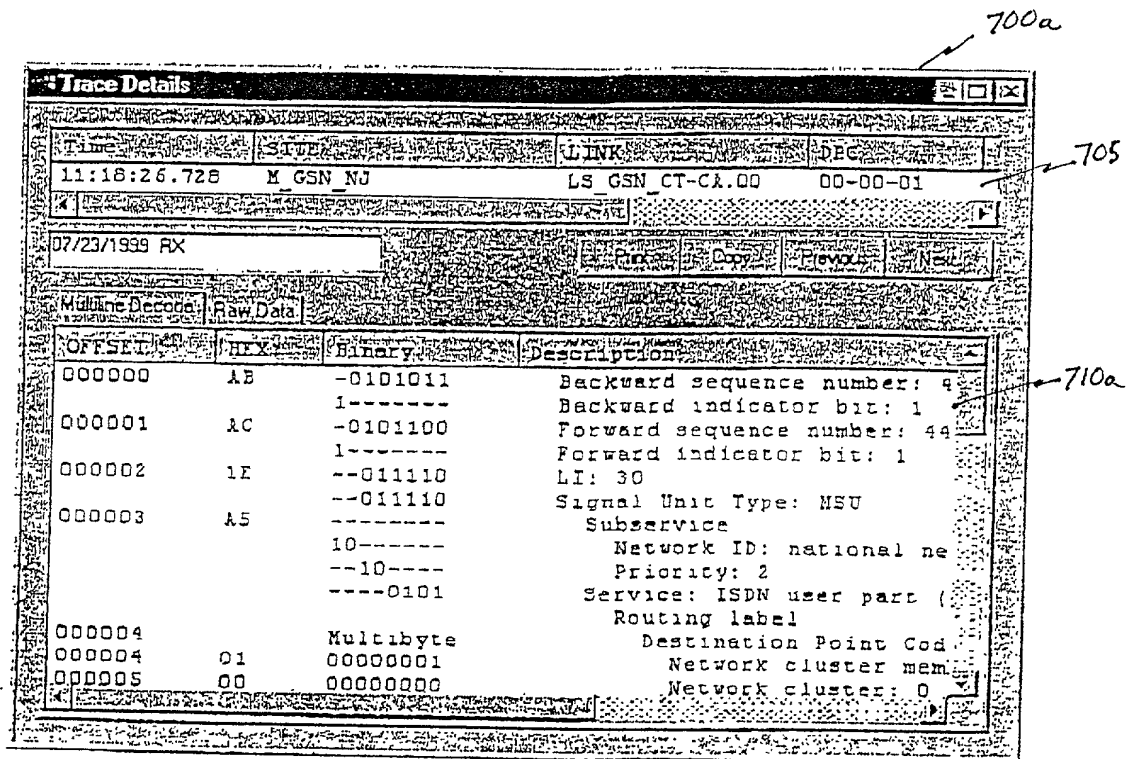


Figure 9-17. Trace Details dialog box showing the Multiline Decode view.

Fig. 7A

Trace Details

11:18:26.728 M GSN NJ LS GSN CT-CA.00 00-00-01

07/23/1999 RX

715a

715b

700b

705

710b

OFFSET	HEX values	ASCII values
000000	AB AC 1E AS 01 00 00 02 00 00	+ . ? . . . . .
000010	00 D2 04 01 00 00 00 00 03 05	. R . . . . .
000020	00 02 80 80 08 81 10 81 00 55	. . . . . U
000030	15 12 02	. . .

Figure 9-18. Trace Details dialog box showing the Raw Data view.

FIG. 7B



## Protocol/Decode

```

* Initialize(loadDefault, bool = true): enum ProtocolDecode::DECODE_ERRORS
* getSiLColumnCount(protocolName : const CString&, headerType : const CString&, colCount : int&): enum ProtocolDecode::DECODE_ERRORS
* getSiLColumnWidth(protocolName : const CString&, headerType : const CString&, colIndex : const int, colWidth : int&): enum ProtocolDecode::DECODE_ERRORS
* getSiLColumnHeader(protocolName : const CString&, headerType : const CString&, colIndex : const int, colName : CString&): enum ProtocolDecode::DECODE_ERRORS
* getSiLColumnHeader(protocolName : const CString&, headerType : const CString&, colIndex : const int, colIndex : LV_COLUMN&): enum ProtocolDecode::DECODE_ERRORS
* decodeSiLColumnHeader(protocolName : const CString&, headerType : const CString&, colIndex : const int, buffer : CBuffer&, dataOut : CString&): enum ProtocolDecode::DECODE_ERRORS
* getSiMColumnCount(protocolName : const CString&, headerType : const CString&, colCount : int&): enum ProtocolDecode::DECODE_ERRORS
* getSiMColumnWidth(protocolName : const CString&, headerType : const CString&, colIndex : const int, colWidth : int&): enum ProtocolDecode::DECODE_ERRORS
* getSiMColumnHeader(protocolName : const CString&, headerType : const CString&, colIndex : const int, colName : CString&): enum ProtocolDecode::DECODE_ERRORS
* getSiMColumnHeader(protocolName : const CString&, headerType : const CString&, colIndex : const int, colIndex : LV_COLUMN&): enum ProtocolDecode::DECODE_ERRORS
* decodeMColumnHeader(protocolName : const CString&, headerType : const CString&, columnIndex : const int, colIndex : const int, buffer : CBuffer&, dataOut : CString&): enum ProtocolDecode::DECODE_ERRORS
* getMLineCount(protocolName : const CString&, headerType : const CString&, buffer : CBuffer&, lineCount : int&): enum ProtocolDecode::DECODE_ERRORS
* getMColumnCount(protocolName : const CString&, sHeader : CString&, int)
* getMColumnWidth(protocolName : const CString&, headerName : const CString&, headerArray : CString&): enum ProtocolDecode::DECODE_ERRORS
* defineCustomHeader(protocolName : const CString&, headerName : const CString&, headerName : const CString&, headerName : const CString&, headerName : const CString&)
* deleteCustomHeader(protocolName : const CString&, headerName : const CString&)
* ProtocolDecode::AppServices : CObject* : ProtocolDecode

```

Fig. 8